THE CLAIMS

This listing reflects the claims pending in the present application.

1. (Previously Presented) An apparatus for performing correctness checks, the apparatus comprising:

logic configured to receive a first set of instructions;

logic configured to generate an initial instruction schedule and a conditional instruction stream from the first set of instructions, such that the initial instruction schedule is devoid of code sequences comprising correctness check functions and such that code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;

logic configured to evaluate the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences associated with correctness check functions can be inserted into the initial instruction schedule such that a final instruction schedule responsive to the initial instruction schedule would not require a longer run time than the initial instruction schedule; and

logic configured to generate the final instruction schedule responsive to the initial instruction schedule, the conditional instruction stream, and the logic configured to evaluate.

2. (Previously Presented) The apparatus of claim 1, wherein correctness check functions are configured to evaluate at least one of a value, a range of values, and a relationship between values after execution of the corresponding instructions in the initial instruction schedule.

3. (Previously Presented) The apparatus of claim 1, wherein said logic configured to generate an initial instruction schedule and a conditional instruction stream from the first set of instructions is responsive to an input provided to a compiler.

4. (Previously Presented) An apparatus for performing correctness checks, the apparatus comprising:

means for receiving a first set of instructions;

means for generating an initial instruction schedule and a conditional instruction stream from the first set of instructions, such that the initial instruction schedule is devoid of code sequences comprising correctness check functions and such that code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;

means for evaluating the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences associated with correctness check functions can be inserted into the initial instruction schedule such that a final instruction schedule would not require a longer run time than the initial instruction schedule; and

means for inserting said code sequences associated with the correctness check function into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule for accommodating said code sequences.

5. (Previously Presented) The apparatus of claim 4, wherein correctness check functions are configured to evaluate at least one of a value, a range of values; and a relationship between values after execution of the corresponding instructions in the initial instruction schedule.

6. (Previously Presented) A method for performing correctness checks, the method comprising the steps of:

receiving a first set of instructions;

generating an initial instruction schedule and a conditional instruction stream from the first set of instructions, such that the initial instruction schedule is devoid of code sequences comprising correctness check functions and such that code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;

evaluating the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences from the conditional instruction stream and associated with correctness check functions can be inserted into the initial instruction schedule such that a final instruction schedule would not require a longer run time than the initial instruction schedule; and

inserting said code sequences associated with the correctness check function into the spare instruction slots if enough spare instruction slots exist in the initial instruction schedule for accommodating said code sequences.

7. (Previously Presented) The method of claim 6, wherein evaluating the initial instruction schedule comprises comparing the run time length of one or more spare instruction slots with the run time length of a code sequence associated with a corresponding portion of the initial instruction schedule.

8. (Previously Presented) The method of claim 6, wherein evaluating the initial instruction schedule further comprises discarding code sequences having a run time length greater than the run time of one or more spare instruction slots associated with a corresponding portion of the initial instruction schedule.

9. (Currently Amended) A computer program for performing correctness checks, the computer program being embodied on a computer-readable medium, the computer program comprising:

a first code segment configured to receive a set of instructions;

a second code segment configured to generate an initial instruction schedule and a conditional instruction stream from the set of instructions, such that the initial instruction schedule is devoid of code sequences comprising correctness check functions and such that the code sequences of the conditional instruction stream are associated with a corresponding set of one or more instructions in the initial instruction schedule;

a third code segment configured to evaluate the initial instruction schedule to determine whether the initial instruction schedule includes spare instruction slots into which said code sequences associated with the correctness check function can be inserted into the initial instruction schedule such that a final instruction schedule would not require a longer run time than the initial instruction schedule; and

a fourth code segment configured to insert code sequences associated with the correctness check function into the spare instruction slots when sufficient spare instruction slots exist in the initial instruction schedule to accommodate said code sequences.

10. (Previously Presented) The computer program of claim 9, wherein said second code segment generates a conditional instruction stream comprising correctness check functions that evaluate at least one of a value, a range of values, and a relationship between values after execution of corresponding instructions in the initial instruction schedule.

11. (Previously Presented) The apparatus of claim 1, wherein said logic configured to evaluate the initial instruction schedule discards code sequences within the conditional instruction stream that if inserted into a final instruction schedule would result in a final instruction schedule with a run time greater than a run time of the initial instruction schedule.

12. (Previously Presented) The apparatus of claim 1, wherein said logic configured to evaluate the initial instruction schedule identifies code sequences within the conditional instruction stream for insertion into the initial instruction schedule.

13. (Previously Presented) The apparatus of claim 12, wherein said logic configured to evaluate the initial instruction schedule identifies code sequences having a length that exceeds the length of a corresponding set of one or more spare instruction slots in the initial instruction schedule.

14. (Previously Presented) The apparatus of claim 1, wherein said logic configured to generate the final instruction schedule inserts code sequences associated with correctness check functions into spare instruction slots of the initial instruction schedule.

15. (Previously Presented) The apparatus of claim 4, wherein said means for generating an initial instruction schedule and a conditional instruction stream from the first set of instructions is responsive to an input provided to a compiler.